Theoretical Work - General Approach

Many optimization problems, both of practical and theoretical importance, are NP-hard. For NP-hard optimization problems, there is no hope of designing efficient (i.e., polynomial-time) algorithms giving the exact solution. One way of dealing with NP-Hard optimization problems is to find polynomial-time approximation algorithms. These algorithms are guaranteed to be fast and to return a near-optimal answer. A typical result is a polynomial-time algorithm for the traveling salesman problem and a proof that the output of the algorithm has value at most 50% more than the optimum.

Besides having a worst-case guarantee, approximation algorithms have two other attractive features. First, often, on real world problems, the approximation algorithms perform much better than the worst-case ratio proved. Sometimes, approximation algorithms are the best performers in practice. Second, to prove a better performance guarantee often requires one to obtain a better understanding of the combinatorial structure of the problem.

1 Scheduling Issues

We propose investigating the algorithmic aspects of assigning jobs to compute nodes together with the scheduling the transfer of files between the cache of the processing and the network attached storage, with the objectives of minimizing makespan and/or communication costs.

There are several variants of these problem. In one, we are given a set A of jobs (applications to be run), each with an access pattern, represented by a sequence $R = (r_1, r_2, \ldots, r_m)$, where each r_i indicates one access to a particular file. Note that the files referenced by different r_i's are possibly the same, and could be on the cache or on the disk. The *cost* is defined as the to-be-evicted file size multiplied by its access frequency after the current processing position in the reference sequence. The *gain* is defined as the to-be-cached file size multiplied by its access frequency after the current fetch position in the reference sequence. Since cache throughput is typically orders of magnitude higher than disks (i.e. O(10GB/s) vs. O(100MB/s)), we simplify the problem statement by ignoring the time of transferring data between the processor and the cache. Similarly, when the file is swapped between cache and disks, only the disk throughput is counted. Our goal is to minimize the total I/O cost of the disk by determining whether the accessed files should be placed in the cache.

A fast heuristic (no performance guarantee) for this problem has been proposed by [7], a paper among whose co-authors are the PI and the PhD advisee of the co-PI. Exact fast algorithms are very unlikely to exist since the problem of finding optimal caching on multiple-disk is proved to be NP-hard [2]. A simpler problem on a single-disk setup has a polynomial solution [1]. An approximation algorithm for multiple disks was proposed in [4] with the restriction that each file size should be the same, which limits its use in practice. In fact, at small scale (e.g. each node has 15 files to access), a brute-force solution with dynamic programming is viable, with the same idea of the classical problem of traveling salesman problem (TSP) [3] with time complexity exponential in the number of files. However, in real applications the number of accessed files could be as large as 10,000, which makes the dynamic programming approach feasible. We propose investigating approximation algorithms for this problem, generalizing to arbitrary file sizes, and possibly improving [4]. The heuristics based on these algorithms can be used in the context of [7], with the potential of improving makespan.

More complex models require assigning jobs to compute nodes as well, taking into account the bounds on computing power of each node.

2 Virtual Machine Migration in Data Centers

Virtual machines (VM) are integral to Apache Hadoop and for the purpose of reliability and load balancing, VMs migration is employed. We propose studying the algorithmic aspects of making the migration cost-effective. Various optimization problems have been formulated, and here we present just one. Originally from [6], we are given a set $V = (V_1, V_2, ..., V_n)$ of VMs of which the first k must be re-assigned, for example due to changes in the availability of physical machines (PM). The remaining $n-k$ VMs are already assigned to PMs. We assume that we know, for each VM, its load, meaning the amount of resources it requires from a PM. PMs each have a capacity. A set $P = (P_1, P_2, ..., P_m)$ of physical machines is also given, together with a function $D(P_k, P_l)$ which represents the distance between two physical machines, and which we assume to be a semimetric, function (that is, obeying the triangle inequality). A traffic demand function $W(V_i, V_j)$ gives the interaction between two VMs (we do not assume this function to be a metric).

A feasible solution consists of assigning the first k VMs to PMs subject to the constraint that the sum of the loads of VMs assigned to a PM does not exceed the capacity of the PM. Let $f(V_i)$ be the PM assigned to VM V_i. The objective is to minimize the data network traffic, expressed as $\sum_{1 \leq i < j \leq n} W(V_i, V_j) D(f(V_i), f(V_j))$.

This problem is NP-Hard even when there are no capacity constraints and all the distances are equal to 1, in which case it becomes the classic Multiway Cut problem [5]. We propose investigating approximation algorithms for the capacited 0-Extension problem we described, as well as practical heuristics based on the ideas developed for the approximation algorithm. These heurstics will then be compared to those of [6].

References

[1] Susanne Albers, Naveen Garg, and Stefano Leonardi. Minimizing stall time in single and parallel disk systems. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 454–462, 1998.

[2] Christoph Ambühl and Birgitta Weber. Parallel prefetching and caching is hard. In *STACS*, pages 211–221, 2004.

[3] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9(1):61–63, January 1962.

[4] Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. A study of integrated prefetching and caching strategies. *SIGMETRICS Perform. Eval. Rev.*, 23(1):188–197, May 1995.

[5] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing*, 23:864–894, 1994. Preliminary version in STOC 1992. An extended abstract was first announced in 1983.

[6] V. Shrivastava, P. Zerfos, Kang-Won Lee, H. Jamjoom, Yew-Huey Liu, and S. Banerjee. Application-aware virtual machine migration in data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 66–70, April 2011.

[7] Dongfang Zhao, Kan Qiao, and Ioan Raicu. Hycache+: Towards scalable high-performance caching middleware for parallel file systems. In *IEEE/ACM CCGrid '14*, 2014.